



实验6 RISC-V汇编器与模拟器实验

1



- 一、实验目的
- 二、实验原理与实验内容
- 三、实验要求
- 四、实验步骤
- 五、思考与探索





一、实验目的



- 学习RISC-V的RV32I指令集，熟悉其指令格式、汇编指令助记符，掌握机器指令编码方法；
- 学习RV32I汇编程序设计，学会使用RISC-V交叉编译器、汇编器，将高级语言程序翻译成汇编语言程序，进而翻译成二进制文件；
- 了解使用RISC-V的模拟器运行程序的方法。





二、实验内容与原理



- **实验内容：**首先**安装**一个RISC-V汇编器与模拟器，然后学会**使用**它对RISC-V汇编语言程序进行**汇编与模拟运行**。学会使用在线编译器工具Compiler Explorer将高级语言程序转换为RISC-V汇编语言程序。



1、计算机的三种语言



- **(1) 机器语言：**用机器指令编写程序，代码语言
 - **机器指令：**能被计算机硬件识别并直接执行的0、1代码串。
- **(2) 汇编语言：**用汇编指令编写程序，符号语言
 - **汇编指令：**用助记符来表示机器指令
 - **汇编器：**将汇编语言程序翻译成机器语言程序的软件
- **(3) 高级语言：**用面向用户的自然语言编写程序，符号语言，可读性好，编程效率高，可移植性好
 - **编译器：**将高级语言程序翻译成二进制机器语言程序的软件



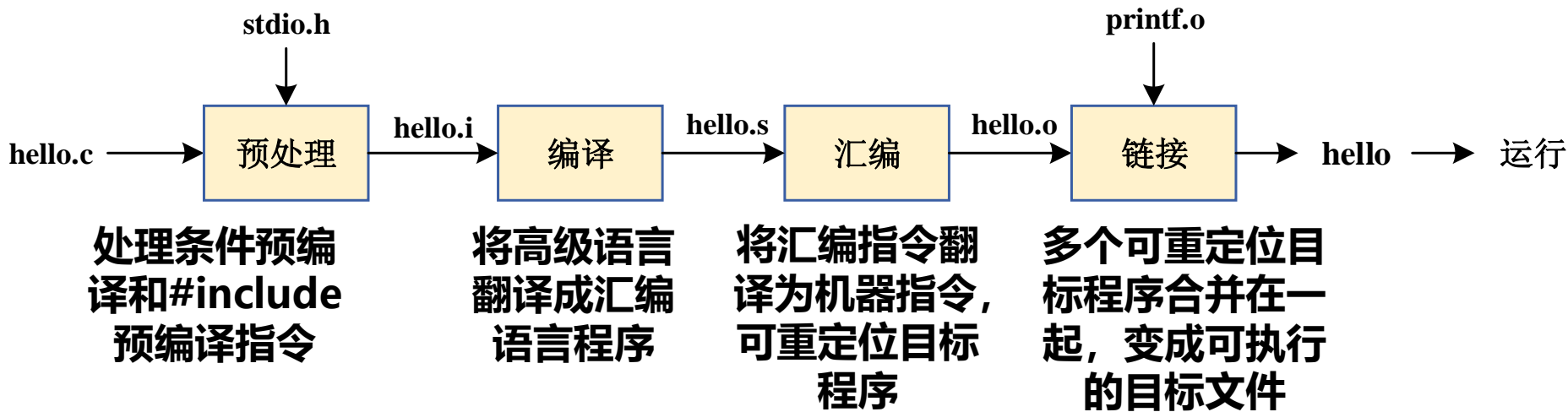
2、编译过程

5



```
#include <stdio.h>
int main() {
    printf("hello world\n");
    return 0;
}
```

■ 高级语言程序→可执行的二进制文件
(机器语言程序)：经过4个阶段





3、RISC-V汇编语言程序

6



■ 例：编写汇编语言程序，并将其翻译成机器语言程序

■ 1) 编辑汇编源程序

■ 输入汇编程序，保存为test.s：用
gedit test.s命令或者任何一个文本
编辑器

■ 程序功能：计算 $(-3) * 8 + 1000 - 7000$

■ 对应的C程序片段是：

```
int a,b,c,w;  
a = -3;          b = 1000;          c = 7000;  
w = 8*a+b-c;
```

```
main:  
li    x5, -3      #-3→x5  
li    x6, 1000    #1000→x6  
li    x7, 7000    #7000→x7  
slli  x8, x5, 3    #x5 << 3→x8  
add   x8, x8, x6   #x8+x6→x8  
sub   x8, x8, x7   #x8-x7→x8  
jr    ra
```



三、实验要求



1. 安装好Visual Studio Code及RISC-V相关的两个插件。用test.s (实验指导书P445) 文件进行汇编器的测试。
2. 将下面一段汇编语言程序acc.s, 借助汇编器, 将其翻译成机器语言程序; 分析其功能; 分析j L1是用什么指令实现的?

main:

add t0,x0, x0

add t1,x0, x0

addi t2,x0, 10

L1: lw t3,0x40(t1)

add t0,t0,t3

addi t1,t1,4

addi t2,t2,-1

beq t2,x0, L2

j L1

L2: sw t0, 0x80(x0)



三、实验要求

3. move.s:

- **子程序BankMove:** 将主存一个地址连续的数据块（数组）复制到主存另一个区域，3个入口参数：
 - a0: 源数据区域的首地址
 - a1: 目标数据区域的首地址
 - a2: 复制的数据个数（数组长度）
- **主程序main:** 调用BankMove, 从内存区域30H复制10个数据到60H

(1)请对上述程序进行汇编;

(2)说明子程序调用和返回的具体实现过程: 解析子程序调用jal和返回指令jr的机器代码, 计算它们的转移地址, 写出指令执行的具体操作与结果, 指出跳转目标地址的指令。

BankMove:

```
add    t0, a0, zero;    #t0=源数据区域首址
add    t1, a1, zero;    #t1=目的数据区域首址
add    t2, a2, zero;    #t2=数据块长度
L1: lw  t3, 0(t0);       #t3=取出数据
sw      t3, 0(t1);       #存数据
addi   t0, t0, 4;        #移动源数据区指针
addi   t1, t1, 4;        #移动目的数据区指针
addi   t2, t2, -1;       #计数值-1
bne    t2, zero, L1;     #计数值≠0, 则没有复制完, 转循环体首部
jr      ra               #复制完成, 则子程序返回
```

main:

```
addi   a0, zero, 0x30;   #a0=0000_0030H, 源数据区域首址
addi   a1, zero, 0x60;   #a1=0000_0060H, 目的数据区域首址
addi   a2, zero, 10;     #a2=0000_000AH, 复制的数据个数
jal    BankMove          #子程序调用
```




三、实验要求

4. 对于下面一段C语言程序sum.c, 请动手编写对应的RV32I汇编程序sum.s, 并进行汇编。

```
int sum(int n)
{
    int i,s=0;
    for(i=0;i<=n;i++)
        s += i;
    return(s);
}
```

```
int main()
{
    int x=100;    int y;
    y = sum(x);
    return 0;
}
```

5. 使用在线编译器工具Compiler Explorer (<https://godbolt.org/>) , 直接将上面的C程序sum.c转换为汇编语言程序, 与自己编写的汇编语言程序对比, 有什么不同? 请进行分析。





四、实验步骤

10



1. 下载并安装Visual Studio Code（已上传到QQ群）及RISC-V相关的两个插件。
2. 用test.s文件进行汇编器的测试。
3. 将指定的汇编语言程序拷贝到acc.s文件，借助汇编器，对其汇编，分析其功能。



四、实验步骤



6. 对move.s汇编程序进行**汇编**，查看子程序调用与返回指令的机器指令代码，分析具体的**字段编码**，计算其**偏移量与目标地址**，得出结论。
7. 编写sum.s的汇编语言程序，**使用汇编器对其汇编**。
8. 使用在线编译器工具Compiler Explorer (<https://godbolt.org/>)，直接将上面的C程序sum.c转换为汇编语言程序，与自己编写的汇编语言程序进行**对比与分析**。





五、思考与探索（至少完成1道）

12



1. `acc.s`程序在功能不变的情况下，可以优化得更简短吗？如果可以，说明你的方法。
2. `move.s`程序中，`BankMove`子程序复制的数据是字节、半字还是字？你是如何判断的？它为何不直接使用`a0~a2`完成复制，而是要将其装入`t0~t2`后再进行处理？
3. 写出调用`BankMove`子程序，复制从内存单元`1000 0000H`到`1000 100H`的20个字数据的主程序。



五、思考与探索（至少完成1道）

13



5. 仔细分析你的sum.s程序，说明你是如何保证边界条件满足的？
6. 华为公司设计与生产了海思麒麟、鲲鹏及昇腾处理器芯片，请查找资料，说明它们是基于什么指令集架构的处理器？它们之间有何区别？对应的指令集架构与RISC-V有何异同点？
7. 查阅文献或者官网资料，了解国产龙芯处理器的指令集架构和应用领域，与RISC-V指令格式作对比。
8. 谈谈你在实验中碰到了哪些问题？又是如何解决的？

